New iMesh paper outline

March 9, 2008

1 Introduction

- Motivation: provide unstructured meshing support for high-performance scientific computing apps, especially SciDAC apps.
- "Support", in this context, essentially means useful for real apps programmers:
  1. general purpose
  2. efficient
  3. support for multiple approaches to same tasks (flexibility)
  4. plug and play interoperability, especially for services
- Intro will also probably include a description of some application and how iMesh would fit in. I'm thinking something simple, like Lori's FE example, because it'll be relatively easy to explain to our audience, but I'm happy to entertain other suggestions. As in, maybe we should talk about a complex app, where the bigger payoffs lie?

2 Design Principles

This section is meant to lay out the guiding principles that we followed to try and accomplish the goals given in the intro. Basically, each of the points in the enumeration above is supported in the interface design by several recurring bases for decision making. In addition to laying those out, I intend to give examples of how that influenced interface design.

1. Completeness. Duh.

2. Run-time efficiency. Low overhead (implying relatively few calls, especially for common tasks) and fast access to data. Examples: single vs. array access, the decisions about how to support mesh shape.

3. Ease of use. A relatively compact interface, with common things easy, even if that makes uncommon things harder. Good example of the former: typed tags. Prime example of the latter: mesh shape.

4. Flexibility. Different apps may want to express the same ideas in different ways. A classic example here is the ways in which sets and tags can be used to represent the same data. Another is (again) single vs. array access.

5. Interoperability. Key to being able to leverage implementation and service development to get plug-and-play services. Requires data structure and language neutrality, as well as component architecture.

3 Data Model

More or less the same as the existing section, but with tie-ins to the design decision section.

4 Interface Functionality

Again, much the same as an existing section. Need to add mesh shape and second adjacencies, as well as cleaning up some things.

5 Examples

This section will replace the current back end of the paper.

Examples I have in mind to discuss in some detail, both as usage examples and to discuss efficiency issues, etc:

- Swapping service
- Mesquite
- Lori's FE example

Then I'd like to be able to say a few words about some of the more complex examples that use iMesh, including things like SLAC (Tim) and fusion (RPI). Once we've got a consensus on which examples and roughly the level of detail we're after, I'll assign homework.

A Specific Function Example

An example of a specific function, as written in the C API, as called from Fortran, as specified by SIDL, and as called from at least one language using Babel.